1. **(15 points)** *Computationally, a vector is simply a list of numbers. We may represent an n-dimensional vector $\vec{a}$ as a list of n coordinates $(a_1, a_2, a_3, \ldots, a_n)$.*

   (a) **(10 points)** *Write an algorithm, using only simple computational steps, to compute the dot product of the vectors $\vec{a}$ and $\vec{b}$. Recall that a dot product of two vectors is the sum of the coordinatewise products, e.g.* $(5,3,1,-2)\cdot(-1,0,4,3) = 5\cdot -1 + 3\cdot 0 + 1\cdot 4 + (-2)\cdot 3 = -7$.

   Input: sequences $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$
   Output: number c
   set c **to** 0;
   set i **to** 0;
   while $i \leq n$ **do**
   | **set** c **to** $c + a_i b_i$;
   | **set** i **to** $i + 1$;
   return determined value c;

   (b) **(5 points)** *Justify and state a good asymptotic bound in big-O notation on the number of steps taken by your algorithm.*

   The innermost set of instructions (adding a product to c) takes constant time, or $O(1)$ time. However, this instruction is repeated when i is 1, when i is 2, when i is 3, and so forth up to when i is n, so the time taken by performing this set of instructions n times is $n \cdot O(1)$, or the linear time $O(n)$.

2. **(10 points)** *Find the closed form of the recurrence relation given by initial conditions $a_0 = 5$, $a_1 = 0$, and $a_n = 2a_{n-1} + 24a_{n-2}$ for $n \geq 2$.*

   Letting $a_n = \lambda^n$ yields the equation $\lambda^n = 2\lambda^{n-1} + 24\lambda^{n-2}$; dividing by λ^{n-2} gives $\lambda^2 = 2\lambda + 24$, which has solutions 6 and -4, so $a_n = 6^n$ and $a_n = (-4)^n$ are both solutions to the recurrence (but not to the initial conditions; thus the general solution of the recurrence is $a_n = k\cdot 6^n + \ell(-4)^n$. To satisfy the initial conditions, it must be the case that $5 = k\cdot 6^0 + \ell(-4)^0$ and $0 = k \cdot 6^1 + \ell(-4)^1$; the solution of this pair of simultaneous equations is $k = 2$, $\ell = 3$, so the closed form for a_n is $2 \cdot 6^n + 3(-4)^n$.

3. **(10 points)** *Consider the following algorithm performed on a sequence of numbers $a_1, a_2, \ldots, a_n$.*

   (1) Let $i = 1$.
   (2) Let $q = i$ and let $j = q + 1$.
   (3) If $a_j < a_q$, then let $q = j$.
   (4) Increment j.
   (5) If $j \leq n$, then return to step (3).
   (6) Swap the values of a_i and a_q (if $i = q$, do nothing).
   (7) Increment i.
   (8) If $i < n$, then return to step (2).

   (a) **(4 points)** *Walk through the algorithm's procedure when performed on the 5-term sequence $(4, 8, 1, 10, 2)$. What does this algorithm seem to do?*

   Steps 2–5 probe each number from i to n, setting q equal to whichever index has the smallest associated value a_q. So, for example, in the first step, when $i = 1$, q would be set to 3, since a_3 is the smallest element of $a_1, \ldots, a_5$. Then that would be swapped to position a_1, so the first time we reach step 7, the sequence would have been modified to $(1, 8, 4, 10, 2)$.

On the second passthrough, when $i = 2$, we would probe from a_2 to a_5 looking for the smallest element; now q would be set to 5, since a_5 is small. So we would swap that to position 2, yielding the sequence $(1, 2, 4, 10, 8)$.

When $i = 3$, we probe a_3, a_4, and a_5 for the smallest; now it is a_3, so we would do nothing.

When $i = 4$, we look at a_4 and a_5; a_5 is smaller, so it is swapped with a_4 to yield $(1, 2, 4, 8, 10)$.

The apparent result of this procedure is to sort our sequence. This is indeed the function of this algorithm, which is known as selection sort. It's a particularly useful sort if our data is for some reason "immobile", since it only swaps when it knows exactly where a number should go, but as we shall see in the next part of this problem, it is not terribly efficient in other ways.

(b) **(6 points)** *Give a big-O estimate of the number of operations, in terms of n, which this algorithm takes to perform its task.*

For each value of i, steps 2–5 will be performed $n - i$ times, since the procedure looks at all values between $i + 1$ and n when seeking the smallest index. The entirety of the procedure will be performed $n - 1$ times: once with $i = 1$, once with $i = 2$, and so forth up to $i = n - 1$, so the total number of cycles through steps 2–5 will be

$$(n-1) + (n-2) + (n-3) + \cdots + 2 + 1 = \frac{n(n-1)}{2} = O(n^2)$$

4. **(12 points)** *Find the particular solution to the recurrence relation $b_n = 4b_{n-1} + 21b_{n-2} - 25 \cdot 2^n$ with initial conditions $b_0 = 1$ and $b_1 = 27$.*

We shall start by finding a solution to the inhomogeneous equation here, and then shall combine it with the homogeneous general solution above; finally we will plug in the initial values to get the constants in the specific solution.

Since the inhomogeneous part is a multiple of 2^n, one solution is likewise a multiple of 2^n: let $b_n = c \cdot 2^n$ be a solution to the inhomogeneous equation. Then:

$$\begin{aligned} c \cdot 2^n &= 4c \cdot 2^{n-1} + 21c \cdot 2^{n-2} - 25 \cdot 2^n \\ c \cdot 2^2 &= 4c \cdot 2^1 + 21c - 25 \cdot 2^2 \\ (4 - 8 - 21)c &= -100 \\ c &= 4 \end{aligned}$$

so one solution to the inhomogeneous equation is $b_n = 4 \cdot 2^n$; note this does not match our initial conditions, however. The *general* solution to the inhomogeneous equation is then $b_n = k \cdot 7^n + \ell(-3)^n + 4 \cdot 2^n$, integrating the general terms from part (a). Using the initial conditions, we can find the particular solution:

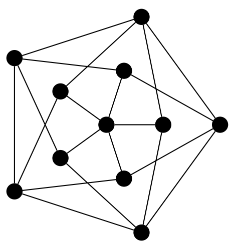
$$\begin{cases} 1 = b_0 = k \cdot 7^0 + \ell(-3)^0 + 4 \cdot 2^0 \\ 27 = b_1 = k \cdot 7^1 + \ell(-3)^1 + 4 \cdot 2^1 \end{cases}$$

which simplifies to

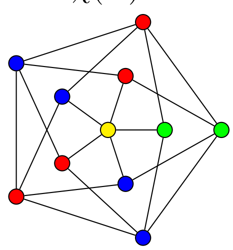
$$\begin{cases} -3 = k + \ell \\ 19 = 7k - 3\ell \end{cases}$$

which can be solved to give $k = 1$ and $\ell = -4$, so the final recurrence is $b_n = 7^n - 4(-3)^n + 4 \cdot 2^n$.

5. **(20 points+5 point bonus)** *Let G be the graph shown below; label vertices as necessary.*



(a) **(10 points)** *Demonstrate via an explicit coloring that $\chi(G) \leq 4$, and give an argument that $\chi(G) > 2$.*



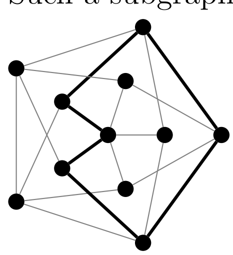
The above picture demonstrates that G is 4-colorable, and thus has chromatic number of no more than 4. In order for G to be 2-colorable, it would have to be bipartite, which is to say, it would need to have no odd cycles as subgraphs, but note that the exterior vertices are joined in a cycle of length 5. Thus G is not bipartite and therefore has chromatic number larger than 2.

(b) **(5 points)** *Is this graph Eulerian? Explain why or why not.*

This graph is not Eulerian, since the 6 interior vertices have odd degree.

(c) **(5 points)** *Demonstrate that this graph has a subgraph isomorphic to C_6.*

Such a subgraph is drawn with heavy lines in the following illustration:



(d) **(5 point bonus)** *Is this graph planar? Either give an explicit planar representation or explain your reasoning.*

It is nonplanar because the five outside vertices, together with the paths of length 2 through the interior, form a subdivision of a K_5.